ASSIGNMENT-4

Assignment Date	29 October 2022
Name	JOHNSON A
Roll Number	420719106017

Question:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

WOKWI Link: https://wokwi.com/projects/347773040334996050

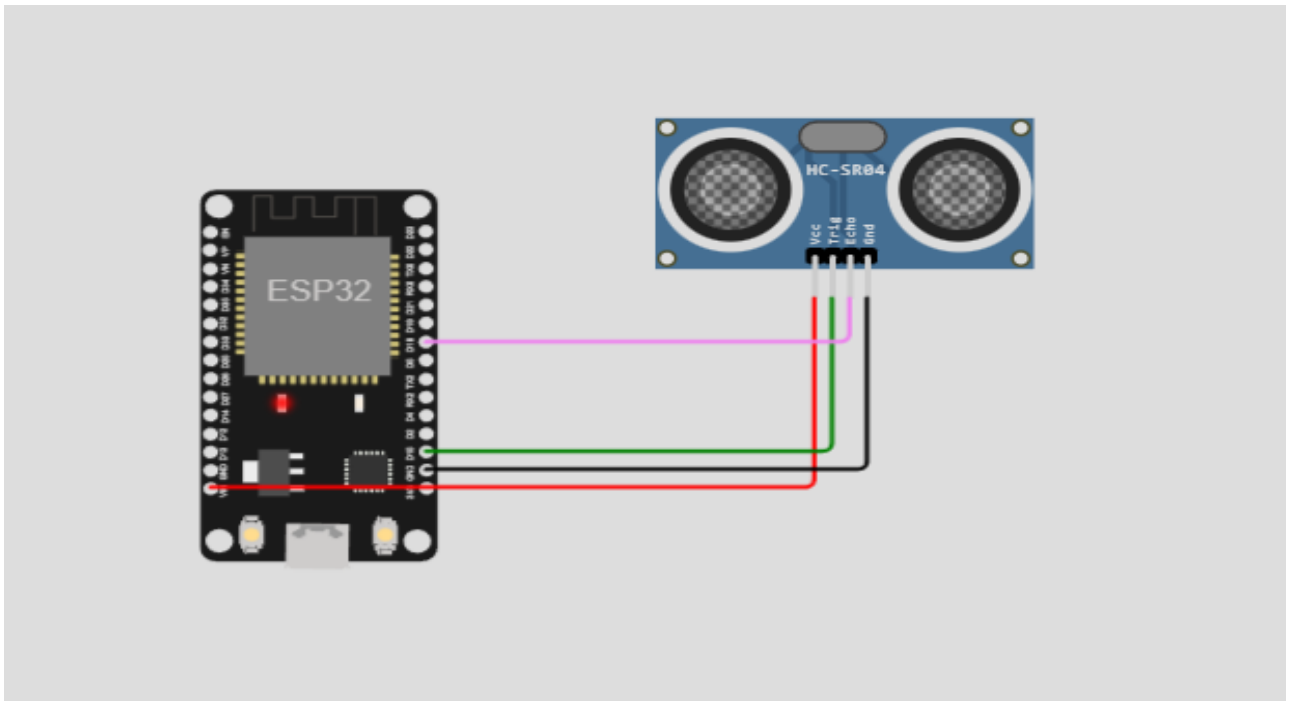
CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define  sound_speed 0.034

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "s2m7ix"//IBM ORGANITION ID
#define DEVICE_TYPE "device_one"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "device_one123"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "device_one123"     //Token
String data3;



//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format
in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS
TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by
passing parameter like server id,portand wificredential

const int trigpin = 15;
const int echopin = 18;



void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(trigpin,OUTPUT);
  pinMode(echopin,INPUT);
  delay(10);
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

    digitalWrite(trigpin, LOW);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
```

```
  digitalWrite(trigpin, LOW);

  long duration = pulseIn(echopin, HIGH);
  float distance = duration *sound_speed/2;
  Serial.print("Distance in cm: ");
  Serial.println(distance);

  if (distance <100){
    delay(1000);
    PublishData(distance);
    delay(1000);
  }
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to Cloud.............................*/

void PublishData(float distance) {
  mqttconnect();  //function call for connecting to ibm
  String object;

  String payload = "{\"Distance\":";
  payload += distance;
  payload += ",\"ALERT!!\":""\"Distance is less than 100cms\"";
  payload += object;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
  }
  else {
    Serial.println("Publish failed");
  }

}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);

    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
```

```
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    Serial.println((char)payload[i]);
    data3 += (char)payload[i];
  }
}
```

CIRCUIT DIAGRAM:



WOKWI OUTPUT:

```
Connecting to ..
WiFi connected
IP address:
10.10.0.2
Reconnecting client to s2m7ix.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance in cm: 96.97
Sending payload: {"Distance":96.97,"ALERT!!":"Distance is less than 100cms"}
Publish ok
Distance in cm: 96.99
Sending payload: {"Distance":96.99,"ALERT!!":"Distance is less than 100cms"}
Publish ok
Distance in cm: 96.99
Sending payload: {"Distance":96.99,"ALERT!!":"Distance is less than 100cms"}
Publish ok
Distance in cm: 276.96
Distance in cm: 276.98
Distance in cm: 276.98
```

IBM CLOUD OUTPUT: